MODELING PITCH TRAJECTORY BY HIERARCHICAL HMM WITH MINIMUM GENERATION ERROR TRAINING

Yi-Jian Wu, Frank Soong

Speech Group, Microsoft Research Asia, Beijing, China

ABSTRACT

A hierarchical pitch model (HPM) was recently proposed to HMM-based speech synthesis. In HPM, pitch trajectory is modeled as an additive combination of hierarchical layers (including state, phone, syllable, etc), and a minimum generation error (MGE) criterion is used to re-estimate model parameters. In this paper, we extend the MGE criterion to a tree-based model clustering process to simultaneously cluster the context-dependent models at all layers, and construct a full MGE training process for HPM training. Experiments were conducted to investigate the effects of HPM with different training criteria and different hierarchical layer combinations. Experimental results show that the full MGE training can significantly improve HPM's ability to predict F0 trajectory in TTS over the ML-based approach on test data. The new HPM also outperforms the conventional state-level HMM in F0 prediction.

Index Terms— Speech synthesis, hidden Markov model, hierarchical pitch model, minimum generation error

1. INTRODUCTION

Hidden Markov model (HMM) based speech synthesis, since it was first proposed [1], has shown its capability to synthesize high quality speech with a flexible model [2]. In HMM, spectrum, pitch and duration are all modeled simultaneously in a unified framework [3], and the parameter trajectories are generated in a maximum probability sense from the HMMs related to the parameter trajectories under a linear constraints between their static and dynamic features.

In conventional HMM-based speech synthesis, a Multi-Space Distribution (MSD) HMM was used for pitch modeling [4], where pitch trajectories are modeled at a state level using multi-state phone HMMs. This pitch modeling method is good at capturing micro prosodic features (e.g., segmental-level perturbation), but difficult to directly characterize long-term pitch patterns, such as pitch accent, phrase level prosody, etc. Although the related prosodic factors are used as context features for context-dependent HMM modeling and clustering, pitch modeling in this inexplicit way still can not well characterize the long-term pitch patterns. In order to solve this over-micro pitch modeling issue, several different methods using hierarchical and/or additive structures had been proposed [5, 6, 7, 8] to capture the pitch patterns related to different prosodic layers.

Previously, we proposed a hierarchical pitch model (HPM) based method [9] to address this issue, where pitch trajectory was decomposed and modeled as an additive combination of hierarchical layers (including state, phone, syllable and word), and a minimum generation error (MGE) criterion was used to re-estimate model parameters for all model layers simultaneously. One issue in this HPM method is that the tree-based clustering of context dependent models for different layers are independent, and then the model size for each layer needs to be tuned manually to achieve a good performance. In this paper, we extend the MGE criterion to the tree-based model clustering process to cluster the context-dependent models at all layers simultaneously, and construct a full MGE training process for HPM training. Under this training process, the model sizes for all HPM layers can be automatically determined.

The rest of this paper is organized as follows. In Section 2, we review the proposed HPM framework, including the model and layer definition, and ML-based training process. In Section 3, we present the details of MGE training for HPM, including MGE-based parameter updating, MGE-based model clustering and a full MGE training process. In Section 4, we describe experiments in evaluating the effects of HPM modeling, and present results. Finally, we give our conclusions and future works in Section 5.

2. HIERARCHICAL PITCH MODELING FRAMEWORK

2.1. Model definition

In our HPM, we assume the acoustic feature trajectories (specifically here, pitch/F0) are generated as a sum of additive components, which is

$$\boldsymbol{o} = \sum_{l=1}^{L} \boldsymbol{o}^{(l)}, \quad \boldsymbol{o}_t = \sum_{l=1}^{L} \boldsymbol{o}_t^{(l)}, \tag{1}$$

where $\boldsymbol{o} = \left[\boldsymbol{o}_1^\top, ..., \boldsymbol{o}_T^\top\right]^\top$ is an acoustic feature vector, $\boldsymbol{o}^{(l)}$ denotes its l-th additive component, and $\boldsymbol{o}_t$ and $\boldsymbol{o}_t^{(l)}$ are the related t-th frame of feature vectors, respectively. Usually, the observation vector consists of both static and dynamic features, i.e., $\boldsymbol{o} = [\boldsymbol{c}^\top, \Delta^{(1)}\boldsymbol{c}^\top, \Delta^{(2)}\boldsymbol{c}^\top]^\top = \boldsymbol{W}\boldsymbol{c}$, where $\boldsymbol{c}$, $\Delta^{(1)}\boldsymbol{c}$ and $\Delta^{(2)}\boldsymbol{c}$ are the static, delta and delta-delta feature vectors, respectively. $\boldsymbol{W}$ is a linear regression matrix which represents the constraints between static and dynamic features.

In training, each additive component in Eq.(1) is modeled by an independent Gaussian distribution model, which is

$$\boldsymbol{o}_t^{(l)} \sim \mathcal{N}\left(\boldsymbol{\mu}_t^{(l)}, \boldsymbol{\Sigma}_t^{(l)}\right) \tag{2}$$

where $\boldsymbol{\mu}_t^{(l)}$ and $\boldsymbol{\Sigma}_t^{(l)}$ are the mean and variance of the corresponding Gaussian distribution at the t-th frame. In synthesis stage, the additive acoustic feature component for each model layer is generated independently [1],

$$\bar{\boldsymbol{c}}_{\boldsymbol{q}_l}^{(l)} = {\boldsymbol{R}_{\boldsymbol{q}_l}^{(l)}}^{-1} \boldsymbol{W}^\top {\boldsymbol{\Sigma}_{\boldsymbol{q}_l}^{(l)}}^{-1} \boldsymbol{\mu}_{\boldsymbol{q}_l}^{(l)}, \tag{3}$$

$$\boldsymbol{R}_{\boldsymbol{q}_l}^{(l)} = \boldsymbol{W}^\top {\boldsymbol{\Sigma}_{\boldsymbol{q}_l}^{(l)}}^{-1} \boldsymbol{W}, \tag{4}$$

where $\boldsymbol{q}_l$ is the state sequence in the l-th layer model, $\boldsymbol{\mu}_{\boldsymbol{q}_l}^{(l)}$ and $\boldsymbol{\Sigma}_{\boldsymbol{q}_l}^{(l)}$ are the mean vector and covariance matrix related to $\boldsymbol{q}_l$. It should be noted that the state sequences are time synchronized for all layers during training and synthesis. Since $\boldsymbol{q}_l$ is embedded in the parameter generation process, we will ignore $\boldsymbol{q}_l$ in the rest of the paper. Finally,

the generated feature vectors of all layers are summed together to form the final output feature vector, i.e.,

$$\bar{\boldsymbol{c}} = \sum_{l=1}^{L} \bar{\boldsymbol{c}}^{(l)}. \tag{5}$$

2.2. Layer definition

The prosodic layers adopted in our HPM framework has four layers including state, phone, syllable and word layers. The state-level and phone-level models capture the pitch dynamics (e.g., segmental pitch perturbation, unvoicing/voicing) in a microscopic manner, and the syllable-level and word-level models model the pitch patterns in a longer range (e.g., pitch accent, phrase tone, etc.)

We adopt a one-state HMM with a single Gaussian for modeling pitch in each layer. The main reason we only use one-state HMM is that the micro prosodic characteristics within each layer can be represented by the pitch models of lower layers, which may not be well modeled by a multi-state HMM in the current layer. In addition, it is easy for implementation to synchronize state sequences between all HPM layers with a unified one-state HMM.

In HMM-based speech synthesis, the MSD-HMM was used to model piece-wise continuous F0 contours, disconnected with unvoiced intervals, where unvoiced/voiced (U/V) property, one of critical part of pitch modeling, can be well characterized. However, the long-term pitch patterns associated with high-level prosodic factors are usually regarded as continuous patterns across a long period, i.e., pitch contours on higher prosodic layers should be treated as continuous. Considering this, we only use MSD-HMM for pitch modeling in the state and phone layers. For the prosodic layers higher than phone layer, we regard the pitch contours as continuous (i.e., all frames are set to be voiced in training and synthesis), and use conventional HMM for modeling.

2.3. ML-based training process

One key issue in HPM training is on how to decompose pitch trajectories into successive prosodic layers hierarchically, and estimate the corresponding model parameters of all layers. In our initial ML-based training process, we use an explicit way to decompose pitch trajectory into each layer, and iteratively update the HPM parameters for each layer. The detailed training process is as follows:

1) Train a conventional state-level pitch model, and segment all training data with the viterbi algorithm. The force aligned state labels are used for parameter generation in all HPM layers, and stay unchanged during the whole ML-based training process.
2) Update the model parameters from the highest layer (i.e., word) to the lowest layer (i.e., state) under the following procedure (shown in Fig. 1):
   a) Generate pitch contours using the pitch models of all layers except for current one;
   b) Calculate the residuals between original and generated pitch trajectories, and use them as training data of current layer;
   c) Train a pitch model for current layer under a standard ML-based HMM training process, which includes context-dependent model training, tree-based clustering and clustered model re-estimation;
3) Iterate Step 2 until reach certain stopping criterion, e.g., maximum number of iterations.

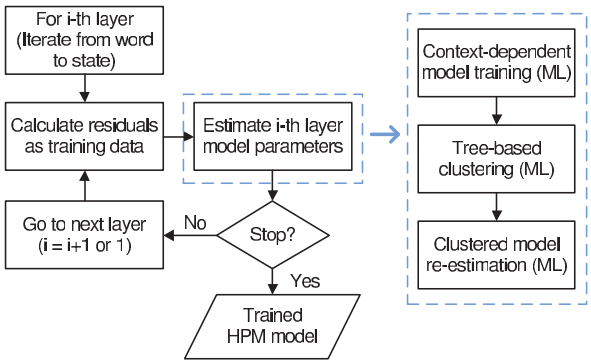

Fig. 1. ML-based hierarchical pitch model training.

There are two issues in the above training process. Firstly, the ML training criterion is not designed for minimizing synthesis errors. Secondly, the pitch models of each layer are estimated while freezing the pitch models of other layers. Due to these two issues, the HPM parameters are not well optimized under the ML-based training process.

3. FULL MGE TRAINING FOR HPM

A minimum generation error (MGE) criterion was proposed [10] for model training in HMM-based speech synthesis, and has demonstrated its effectiveness to improve synthesized speech quality [11]. In this paper, we adopt the MGE criterion in training HPM, including parameter re-estimation and tree-based clustering for context dependent models.

3.1. MGE-based parameter re-estimation

Similar to [10], the generation error for a given feature vector $\boldsymbol{c}$ is defined as Euclidean distance between the original feature vector and generated feature vector, which is

$$e(\boldsymbol{c}, \boldsymbol{\lambda}) = \|\bar{\boldsymbol{c}} - \boldsymbol{c}\|^2 = \left\| \sum_{l=0}^{L} \bar{\boldsymbol{c}}^{(l)} - \boldsymbol{c} \right\|^2, \tag{6}$$

where $\boldsymbol{\lambda}$ denotes the HPM parameters.

The objective of MGE criterion is to optimize the model parameters so as to minimize the total generation (synthesis) errors, i.e,

$$\hat{\lambda} = \arg\min E(\lambda) = \arg\min \sum_{n} e(\boldsymbol{c}_n, \lambda). \tag{7}$$

As a close-form solution for Eq. (7) is mathematically intractable, a probabilistic descent (PD) method was adopted for parameter optimization. In PD method, the update of HMM parameters is

$$\lambda(\tau+1) = \lambda(\tau) - \epsilon_\tau \frac{\partial e(\boldsymbol{c}, \boldsymbol{\lambda})}{\partial \lambda}\bigg|_{\lambda=\lambda_\tau}, \tag{8}$$

where ϵ_τ is the step size to control the convergence speed. Finally, the updates of mean and variance parameters can be formulated as

$$\boldsymbol{\mu}^{(l)}(\tau+1) = \boldsymbol{\mu}^{(l)}(\tau) - 2\epsilon_\tau {\boldsymbol{\Sigma}^{(l)}}^{-1} \boldsymbol{W} {\boldsymbol{R}^{(l)}}^{-1} (\bar{\boldsymbol{c}}_\tau - \boldsymbol{c}_\tau), \tag{9}$$

$$\log \boldsymbol{\Sigma}^{(l)}(\tau+1) = \log \boldsymbol{\Sigma}^{(l)}(\tau) - 2\epsilon_\tau {\boldsymbol{\Sigma}^{(l)}}^{-1} \boldsymbol{W} {\boldsymbol{R}^{(l)}}^{-1} \cdot (\bar{\boldsymbol{c}}_\tau - \boldsymbol{c}_\tau) \left(\boldsymbol{W} \bar{\boldsymbol{c}}_\tau^{(l)} - \boldsymbol{\mu}^{(l)} \right)^\top, \tag{10}$$

where $\boldsymbol{\mu}^{(l)}$ and $\boldsymbol{\Sigma}^{(l)}$ are the mean vector and covariance matrix related to the model sequence $\boldsymbol{q}_l$ of l-th layer for feature vector $\boldsymbol{c}_\tau$.

3.2. MGE-based model clustering

Similar to the MGE-based model clustering for conventional HMM training [12], we apply the MGE criterion to HPM clustering, where the node splitting score is calculated as the reduction of generation errors after splitting, and the HPM parameters of cluster node are re-estimated under MGE criterion after each splitting.

In MGE-based model clustering, the parameter updating rules of MGE criterion are time consuming, which is mainly due to the sample-by-sample updating manner and the calculation of R^{-1}. In order to reduce the computational cost, the parameter updates are simplified, and batch processing is used for parameter updates, i.e.,

$$\lambda_{\tau+1} = \lambda_\tau - \epsilon_\tau \sum_n \boldsymbol{H}^{-1}(\boldsymbol{c}_n, \lambda) \frac{\partial e(\boldsymbol{c}_n, \lambda)}{\partial \lambda}\bigg|_{\lambda=\lambda_\tau} \tag{11}$$

where $\boldsymbol{H}(\boldsymbol{c}_n, \lambda)$ is the Hessian matrix. Here we minimize the total generation error $E(\lambda)$ with respect to

$$\boldsymbol{m}^{(l)} = \left[\boldsymbol{\mu}_1^{(l)\top}, \boldsymbol{\mu}_2^{(l)\top}, \ldots, \boldsymbol{\mu}_{N_l}^{(l)\top}\right]^\top, \tag{12}$$

$$\boldsymbol{U}^{(l)} = \left[\boldsymbol{\Sigma}_1^{(l)-1}, \boldsymbol{\Sigma}_2^{(l)-1}, \ldots, \boldsymbol{\Sigma}_{N_l}^{(l)-1}\right]^\top, \tag{13}$$

where $\boldsymbol{m}^{(l)}$ and $\boldsymbol{U}^{(l)}$ are defined by concatenating the mean vectors and covariance matrices of all unique Gaussian components in the l-th layer model set; $\boldsymbol{\mu}_i^{(l)}$ and $\boldsymbol{\Sigma}_i^{(l)}$ are the mean vector and covariance matrix of the i-th unique Gaussian component of l-th layer, and N_l is the number of Gaussian components in the l-th layer, respectively.

To alleviate the computational cost of Hessian matrix, we use the following pseudo-inverse matrix to approximate the inverse of Hessian matrix for mean and variance, which are

$$\tilde{\boldsymbol{H}}_{\boldsymbol{\mu}}^{-1} = \boldsymbol{W}\boldsymbol{W}^\top, \tag{14}$$

$$\tilde{\boldsymbol{H}}_{\log \boldsymbol{\Sigma}^{(l)}}^{-1} = \boldsymbol{\Sigma}^{(l)-1}\boldsymbol{W}\boldsymbol{W}^\top. \tag{15}$$

Then the updating rules for $\boldsymbol{m}^{(l)}$ and $\boldsymbol{U}^{(l)}$ can be formulated as

$$\boldsymbol{m}^{(l)}(\tau+1) = \boldsymbol{m}^{(l)}(\tau) - 2\epsilon_\tau^{(l)} \sum_n \boldsymbol{S}^{(l)}(\bar{\boldsymbol{o}}_n - \boldsymbol{o}_n), \tag{16}$$

$$\log \boldsymbol{U}^{(l)}(\tau+1) = \log \boldsymbol{U}^{(l)}(\tau) - 2\epsilon_\tau^{(l)} \sum_n \boldsymbol{S}^{(l)}\boldsymbol{\Sigma}^{(l)-1} \cdot (\bar{\boldsymbol{o}}_n - \boldsymbol{o}_n)(\bar{\boldsymbol{o}}_n^{(l)} - \boldsymbol{\mu}^{(l)})^\top, \tag{17}$$

where $\boldsymbol{S}^{(l)}$ is a $3MT \times 3MN_l$ matrix whose elements are 0 or 1 determined according to the optimal model sequence $\boldsymbol{q}_l$ of l-th layer for a feature vector $\boldsymbol{c}_n$.

In addition to simplifying the MGE-based parameter updates to reduce the computational cost, we adopt a method to combine the MGE with ML criterion to select splitting questions for each node. In this method, the ML criterion is firstly used to pre-select a subset of the questions in an efficient way. Then the simplified MGE criterion is applied to select the best splitting question from the subset of pre-selected questions.

As we mentioned in Sec. 1, one of the problems in ML-based clustering is that the model clustering for each layer is independent, and we need to manually set the model size for each layer. In MGE-based model clustering, all the HPMs of all layers are clustered simultaneously, and the tree size of each model layer can be automatically determined. We only need to set the total number of models summed over all HPM layers.

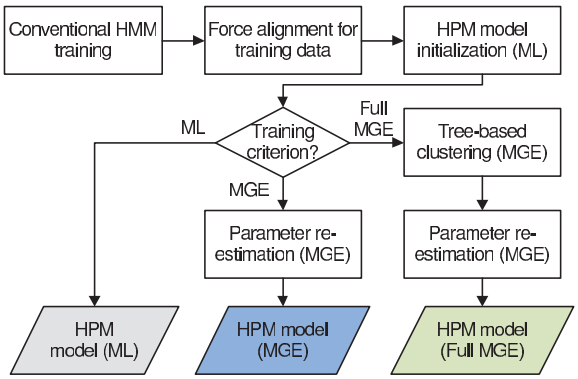

Fig. 2. ML/MGE/Full MGE training process for HPM

3.3. Full MGE training process

With the MGE-based parameter re-estimation and model clustering techniques, we construct a full MGE training process for HPM, which is shown in the right part of Fig. 2. From this training process, the ML criterion is only used to initialize HPM models. After that, the context dependent model clustering and clustered model re-estimation are all based on MGE criterion.

3.4. Discussions

In the HPM training, we apply an alignment refining process to refine state alignments during the MGE-based parameter updating and model clustering. The refining process is a heuristical process to search optimal alignment under the MGE criterion, which is similar to [13]. Previous work in [9] demonstrated that the alignment refining process can improve the MGE training performance for HPMs, and has little impact on the phonetic spectral models.

As we mentioned in Sec. 2, we used the MSD-HMM for pitch modeling for state and phone layers, and use conventional continuous HMMs for modeling of higher layers. Therefore, we need to consider the U/V weights in both state-level and phone-level models for U/V decision in pitch generation. In our current experiments, this problem was avoided by directly using U/V decisions from original pitch trajectory during model training and objective testing. More investigations are needed to find the best way to combine the U/V weights of state-level and phone-level models.

4. EXPERIMENTS

4.1. Experimental conditions

A female English speech database containing 3,910 phonetically balanced sentences was used in our experiments. Sampling rate of recorded speech waveforms was 16kHz, and frame shift was set to 5ms. 2,969 sentences were selected from the database for model training, and the rest of database is used as test data. The acoustic features consist of logF0, 40-th order LSPs and gain. A 5-state left-to-right HMM with no skip was adopted. Minimum description length (MDL) criterion [14] was adopted to determine the number of clustered models in ML-based clustering.

We investigated the effects of three different training processes for HPM training (including ML training, MGE training, and full MGE training, which are shown in Fig. 2), and the effects of different layer combinations in our experiments. The number of iterations for ML-based HPM initialization is set to 2, and the number of MGE training iterations is set to 10. For comparison, we also conducted

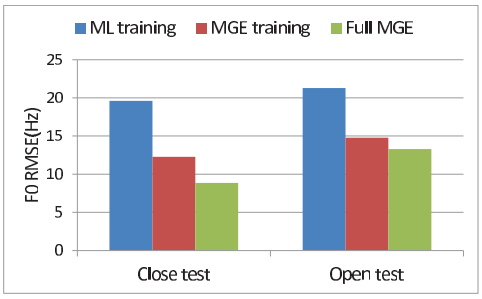

Fig. 3. Effects of different HPM training processes

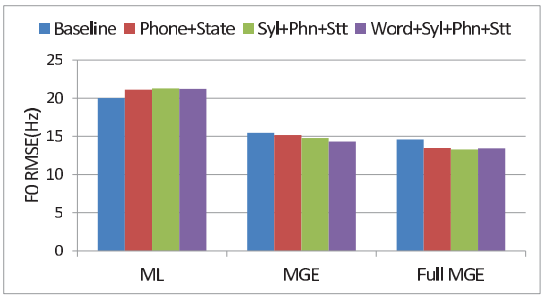

Fig. 4. Comparison of HPMs at different model layers

the ML, MGE and full MGE training for conventional state-level HMMs. The alignment refinement is applied in the MGE-based parameter updating and model clustering process in all experiments. In order to make a fair comparison, the total numbers of model parameters (i.e., number of clustered models) for different training criteria and different layer combinations are set to roughly the same numbers by setting MDL values.

4.2. Experimental results

4.2.1. Effects of different training processes

We used the RMSE between generated and original F0 trajectories as an objective measurement to evaluate the performance of different pitch models. Because the results of the HPMs with different layer combinations are similar, here we only show the results for the HPM with state, phone and syllable layers. Fig. 3 shows the effects of different HPM training processes (ML, MGE and full MGE). It can be seen that the MGE training significantly improved the performance of ML-trained HPMs, and the full MGE training further improved the performance. Comparing to the ML-trained HPMs, the relative reductions of F0 RMSEs after the full MGE training are 55% and 37% on the close and open test, respectively. In addition, it should be noted that the differences of F0 RMSEs between close and open tests become larger after applying the MGE and full MGE training, which indicates the MGE-based parameter updating and full MGE training may over-fit the training data.

4.2.2. Effect of different layer combinations

We compared the performance of the following HPMs and the baseline pitch model:

- Baseline: conventional state-level pitch model
- Phone+State: HPMs at phone and state layers
- Syl+Phn+Stt: HPMs at syllable, phone and state layers
- Word+Syl+Phn+Stt: HPMs at word, syllable, phone and state layers

The open test results of HPMs under ML, MGE and full MGE training are shown in Fig. 4. It can be seen that the performance of ML-trained HPMs are quite similar, and are all worse than the baseline state-level pitch model. After the MGE training, the HPMs outperformed the baseline model, and more layers indicated better performance. After the full MGE training, the performance of HPM models were further improved. However, the HPMs with more layers did not show appreciable performance improvement, which may due to the over-fitting issue of the full MGE training.

5. CONCLUSIONS AND FUTURE WORK

This paper presents a hierarchical pitch model (HPM) method with a full MGE training process, where the MGE criterion has been applied to context-dependent model clustering and clustered model re-estimation. Experimental results showed the proposed HPMs with full MGE training significantly reduced the F0 RMSEs on the test data, compared to the conventional state-level pitch model. Future works include conducting a more detailed subjective listening test and investigating the over-fitting issue in MGE training.

6. REFERENCES

[1] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai, "Speech synthesis from HMMs using dynamic features," in *Proc. of ICASSP*, pp. 389–392, 1996.

[2] Z.-H. Ling, Y.-J. Wu, Y. Wang, L. Qin, and R.-H. Wang, "USTC system for Blizzard Challenge 2006 an improved HMM-based speech synthesis method", in *Blizzard Challenge 2006*.

[3] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. of ICASSP*, pp. 2347–2350, 1999.

[4] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden markov models based on multi-space probability distribution for pitch pattern modeling," in *Proc. of ICASSP*, pp. 229–232, 1999.

[5] S. Sakai, "F0 modeling with multi-layer additive modeling based on a statistical learning technique," in *Proc. of SSW5*, pp. 151–154, 1999.

[6] Y. Qian, H. Liang, and F. K. Soong, "Generating natural F0 trajectory with additive trees," in *Proc. of Interspeech*, pp. 2126–2129, 2008.

[7] C.C. Wang, Z.H. Ling, B.F. Zhang, and L.R. Dai, "Multi-layer F0 modeling for HMM-based speech synthesis," in *Proc. of ISCSLP*, pp. 129–132, 2008.

[8] H. Zen and N. Braunschweiler, "Context-dependent additive log F0 model for HMM-based speech synthesis," in *Proc. of Interspeech*, pp. 2126–2129, 2009.

[9] M. Lei, Y.-J. Wu, F. K. Song, Z.-H. Ling, L.-R. Dai, "A Hierarchical F0 Modeling Method for HMM-based Speech Synthesis," in *Proc. of Interspeech*, pp. 2170–2173, 2010.

[10] Y.-J. Wu and R.H. Wang, "Minimum generation error training for HMM-based speech synthesis," in *Proc. of ICASSP*, pp. 889–892, 2006.

[11] Y.-J. Wu, R.H. Wang, and F. Soong "Full HMM training for minimizing generation error in synthesis," in *Proc. of ICASSP*, pp. 517–520, 2007.

[12] Y.-J. Wu, W. Guo and R.H. Wang, "Minimum generation error criterion for tree-based clustering of context dependent HMMs," in *Proc. of Interspeech*, pp. 2046-2049, 2006.

[13] Y.-J. Wu, L. Qin, and K. Tokuda, "An improved minimum generation error based model adaptation for HMM-based speech synthesis," in *Proc. of Interspeech*, pp. 1787–1790, 2009.

[14] K. Shinoda and T. Watanabe "MDL-based context dependent subword modeling for speech recognition," in *J. Acoust. Soc. Jpn.(E)*, vol. 21, no. 2, pp. 79–86, 2000.